A Framework for Event Log Generation and Knowledge Representation for Process Mining in Healthcare

Roberto Gatta
Istituto di Radiologia
Universitá Cattolica del Sacro Cuore
Rome, Italy

Mauro Vallati
School of Computing and Engineering
University of Huddersfield
Huddersfield, UK

Jacopo Lenkowicz
Istituto di Radiologia
Universitá Cattolica del Sacro Cuore
Rome, Italy

Calogero Casá
Istituto di Radiologia
Universitá Cattolica del Sacro Cuore
Rome, Italy

Francesco Cellini
Polo Scienze Oncologiche
Policlinico Universitario A. Gemelli
Rome, Italy

Andrea Damiani
Istituto di Radiologia
Universitá Cattolica del Sacro Cuore
Rome, Italy

Vincenzo Valentini
Istituto di Radiologia
Universitá Cattolica del Sacro Cuore
Rome, Italy

***Abstract*—Process Mining is of growing importance in the healthcare domain, where the quality of delivered services depends on the suitable and efficient execution of processes encoding the vast amount of clinical knowledge gained via the evidence-based medicine paradigm. In particular, to assess and measure the quality of delivered treatments, there is a strong interest in tools able to perform conformance checking.**

In process mining for the healthcare domain, a number of major challenges are posed by: (i) the complexity of involved data, that refers to patients' aspects such as disease, behaviour, clinical history, psychology, etc; (ii) the availability of data, that come from the heterogeneous, fragmented and scant connected healthcare system; and (iii) the wide range of available standards for communication (DICOM, IHE, etc.) or data representation (ICD9, SNOMED, etc.) purposes.

To effectively perform process mining in the healthcare domain, it is crucial to build event logs capturing all the steps of running processes, which have to be derived by the knowledge stored in the Electronic Health Records. It is therefore crucial to cope with aforementioned data-related challenges.

In this paper, we aim at supporting the exploitation of process mining in the healthcare domain, particularly with regards to conformance checking. We therefore introduce a set of specifically-designed techniques, provided as a suite of software packages written in R. In particular, the suite provides a flexible and agile way to automatically and reliably build Event Log from clinical data sources, and to effectively perform conformance checking.

***Index Terms*—Process Mining, Healthcare, Conformance Checking**

I. Introduction

In the daily clinical practice, the complexity of the biology and the assortment of patients' response to therapies and diseases, forces clinicians to constantly tune, *in itinere*, the strategy of the cures, stretching the notion of *Personalized Medicine* [1], [2]. However, for clear reasons, strategies must be aligned with appropriate Clinical Guidelines, which are probably the most visible expression of *Evidence-Based Medicine* [3]. Recent advances suggest that process mining (PM) techniques may provide a valuable tool for handling such dichotomy, i.e. the need for personalized medicine that follows guidelines expressing evidence-based medicine. For this reason, since the presentation of the process mining manifesto [4], process mining generated very high expectations in the healthcare domain, where the quality of services depends on the effective and efficient execution of processes. Healthcare processes include a sequence of activities for diagnosing, treating, and preventing any disease, in order to improve patients' health. Such processes are supported by both clinical and non-clinical activities, executed by different human resources and can vary between different hospitals [5].

Due to the heterogeneity and sparsity of the data sources, the existence of many dictionaries for data representation with partially overlapped aims (such as ICD9, SNOMED, LOINC), the abundance of legacy solutions, the large number of communication protocols (DICOM, HL7, IHE, etc.), and a growing number of ontologies (uncontrolled in number and with no guarantees on quality)[1], the healthcare domain represents a hard challenge in particular for data acquisition and data representation. Hence, general PM approaches can hardly be straightforwardly applied in healthcare, and that lead to the creation of the Process Mining for Healthcare (PM4HC) discipline, which is currently being supported and

[1]An example of ontology for the healthcare domain can be found at https://bioportal.bioontology.org/

characterised by a number of dedicated initiatives, such as specific courses[2] or consortia.[3]

Existing tools for process mining in the healthcare domain assume the presence of an event log, and do not deal with data extraction, and the subsequent data collection and harmonization. As previously observed, dealing with data is one of the major issues of PM in the healthcare domain. Therefore, existing PM tools can only be used for "offline" analysis, as they are not able to integrate and interface with Electronic Health Records (EHR), and they require a significant preprocessing of available data in order to generate event logs in a format that is suitable for them. One of the pivotal challenges of PM4HC is therefore the design and development of integrated solution able to implement all the pipeline from the data acquisition to the delivery of the PM analysis. This is particularly true for conformance checking, due to the critical role it plays in daily clinical practice, and to the increasing availability of healthcare data. Due to the evidence-based paradigm, and the exposure to legal issues in cases where the clinical practice does not comply with guidelines and protocols, there is a strong interest in tools and methods able to measure/ check/ verify/ alert *how* and *when* the patient's clinical pathways are following one or more desired patterns.

pMineR [6] is the first library designed to cope with all the aspects of process mining in healthcare in the R [7] environment. pMineR, for instance, has been used to generate and compare knowledge graphs extracted by available medical data [8]. It can interact with the vast amount of the available R packages in order to generate graphs, dashboards, provide statistics, exploit different analysis approaches, etc. In pMineR, conformance checking can be performed using a specifically-devised language called Pseudo Workflow Language (PWL): it is a language designed from a multidisciplinary team of clinicians and computer scientists and is aimed at representing clinical workflows (such as clinical guidelines) and exploring how the tracks of a given event log flow through the represented workflow. However, as the other existing approaches for PM4HC, pMineR requires preprocessed event logs to be provided.

In this paper, in order to complete the computation pipeline with a module able to provide a data source connection and automatically extract the event logs, and fill the gap between PM approaches and data sources, we propose *Ste*: an R package designed to be integrated with pMineR and able to automatically extract data from existing clinical data sources (mainly EHRs). To achieve this goal, Ste implements an engine to parse scripts written in the CSL [9] language. Remarkably, CSL can also be used for representing clinical guidelines, enriching the suite of tools available to deal with conformance checking and overcoming the limitations of PWL. In particular, PWL has limited expressive power for encoding temporal aspects of CIGs, and requires Event Logs in order to correctly perform conformance checking.

[2]see, for example https://www.futurelearn.com/courses/process-mining-healthcare

[3]e.g., http://www.processmining4healthcare.org/

Here we demonstrate how Ste can be exploited for data extraction and event log building from an existent EHR, in order to provide suitable data for pMineR. Furthermore, we show how Ste can directly perform conformance checking tasks, bypassing the issues of the PWL language currently used by the pMineR infrastructure. The analysis, performed on real-world data from $12,000$ patients of the Gemelli Hospital, allows us to investigate the capabilities of Ste, and to highlights its strengths and weaknesses.

The remainder of this paper is organised as follows. Firstly, we present the relevant background by introducing related works in the PM4HC area. Then, we briefly describe pMineR and PWL. Subsequently. we describe Ste, and the underlying CSL language, and we demonstrate how they can be used to effectively generate Event Logs, and to perform conformance checking. Finally, we discuss lessons learnt during our analysis, and give conclusions.

II. Process Mining in the Healthcare Domain

Applications of PM4HC include, among others, its exploitation of data from the fields of Oncology [10], and Surgery [11], providing significant insights about the executed tasks and how the clinicians and experts can improve their services.

The first general review of the PM4HC discipline was provided by Roja et al. [12], but recently, more specific reviews have been proposed, for example in oncology [13]. Remarkably, most of the existing reviews focus on conformance checking more than on process discovery.

The automated compliance verification of clinical processes with regards to given guidelines has been well-studied, and thoroughly investigated, in the area of Computer-Interpretable Clinical Guidelines (CIGs). Arden Syntax [14], the most famous language for representing clinical guidelines and now sponsored by the Health Level Seven International organisation, was presented in 1992; Asbru [15] and GLIF [16] were also introduced in the same period, and a mature review of the discipline has been available since 2013 [17].

The process mining manifesto [4] introduced many elements of innovation, in particular with the Machine Learning perspective in dealing with real-world clinical data and processes. For this reason, the partial "overlap" between the PM4HC and CIGs disciplines, and the possible benefits deriving from cross-fertilisation, are points of interest for the scientific community. Beside other relevant aspects, there is currently a growing interest in investigating how CIGs techniques and approaches can be enhanced and extended in the PM4HC framework, in particular for conformance checking.

In terms of available software and tools, with the exception of few commercial solutions (that do not usually support the specific issues of the healthcare domain), there is a relatively scant offer of approaches to cope with PM4HC. A well-known approach is PROM [18], written in JAVA, which provides a GUI and APIs, and is rich of plug-ins to support a large part of the PM tasks. An alternative, that is free to use only for academic purposes, is DISCO [19] which provides a more friendly environment, at the cost of a reduced flexibility. A

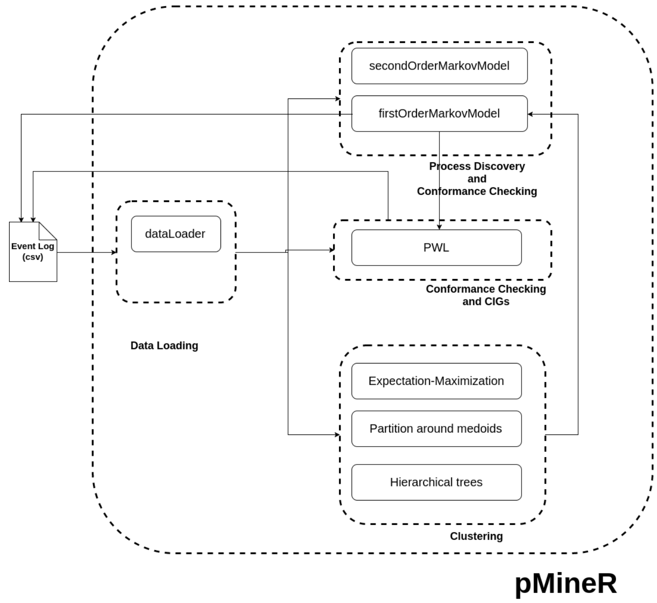

Fig. 1. A schema representing the main modules of pMineR, and their interactions. In this schema, pMineR is loading data directly from an existing Event Log.

PALIA ILS Suite Web Tool was used by Fernández-Llatas et al. [20], and it presented some innovative ideas to explore the PM tasks and issues in the healthcare domain.

III. pMineR

pMineR[4], available in the most updated version on github (https://github.com/kbolab/pMineR), is a free and open source R package available on CRAN. The architecture of pMineR is shown in Figure 1, and is based on a number of independent modules, to cope with:

- Process discovery and conformance checking, following the traditional PM perspective. Two main algorithms are currently implemented for process discovery: one algorithm is based on first order Markov models, while the other implements the Careflow Mining technique [21].
- Process discovery and conformance checking, taking into account the peculiarities of CIGs. For this purpose, PWL is exploited.
- Clustering is currently supported through expectation-maximization [22], partition around medoids [23], and hierarchical trees.
- Data loading.

The architecture is modular, and has been designed in order to be easy to extend. Each "module" is devoted to a specific task, and can interact with the other modules only via a standardised interface, composed by public methods. For this reason, each module can be interchanged without affecting the rest of the framework.

Importantly, when dealing with conformance checking while taking into account CIGs, pMineR introduces and exploits a specific language, called PWL, to represent simple clinical guidelines and requires, in input, the presence of a given Event Log in the form of a csv file.

[4]http://www.pminer.info

A. PWL

Pseudo Workflow Language (PWL) is a language designed to represent simple clinical workflows. It allows to represent workflows under the form of transition graphs, and to encode them as XML. In fact, a script in PWL is an XML file composed by three main constructs:

- Events: in a PWL execution, the event log is processed in a first-in first-out way. This construct refers to the event that is currently under analysis, and can include attributes and properties of the event.
- Nodes: this construct is used to represent states' activation status (active or off) during the execution. A state, for instance, can be used to represent the condition of the patient under analysis. In order to represent complex situations, more nodes can be active at the same time.
- Triggers: they are structured in three sections: *condition*, *set* and *unset*: triggers are automatically fired when the *condition* is satisfied. Their execution results in the *set* and *unset* sections, respectively, to activate or deactivate specified nodes. The condition is a composed expression (`AND|OR|NOT`) that can check the state of the nodes, the current event considered, or some of the attributes of one or more events.

A set of scripts and XML files can be found at http://helios.hud.ac.uk/scommv/storage/bpm-example.zip. Hereinafter, we will refer to this archive as the example archive, and we will exploit it for providing examples of scripts and constructs encoded using the considered languages and approaches. The included **RTCHR.flusso.medio.xml** gives an example of a PWL script, and the corresponding computation is shown in Figure 2. It is worth emphasising that PWL can be also used for conformance checking purposes, and to manipulate event logs in order to update or modify transitions between nodes. However, it does rely on the presence of a valid and noise-free event log.

IV. Ste and CSL

Ste implements and exploits the CSL language [9], which was aimed at bridging the gap between the clinical data stored in the EHR and Decision Support Systems (DSS). Ste provides a way to link RDMBS (in the current version MySQL and PostgreSQL), and it is also able to manipulate the data and present to the user inferenced data, in order reduce the gap between different levels of granularity in EHRs, and meet the needs of a range of final users (either human or DSS software agents). From the point of view of the CIGs discipline, Ste can help to cope with the well-known "curly brackets problem" [24], [25].

CSL is based on a two-layers architecture: at the lower layer it uses a mapping language called LLL (Lower Layer Language). The aim of LLL is to map entities and relations of the underlying ER schema to an appropriate structure composed following the object-oriented paradigm. In fact,

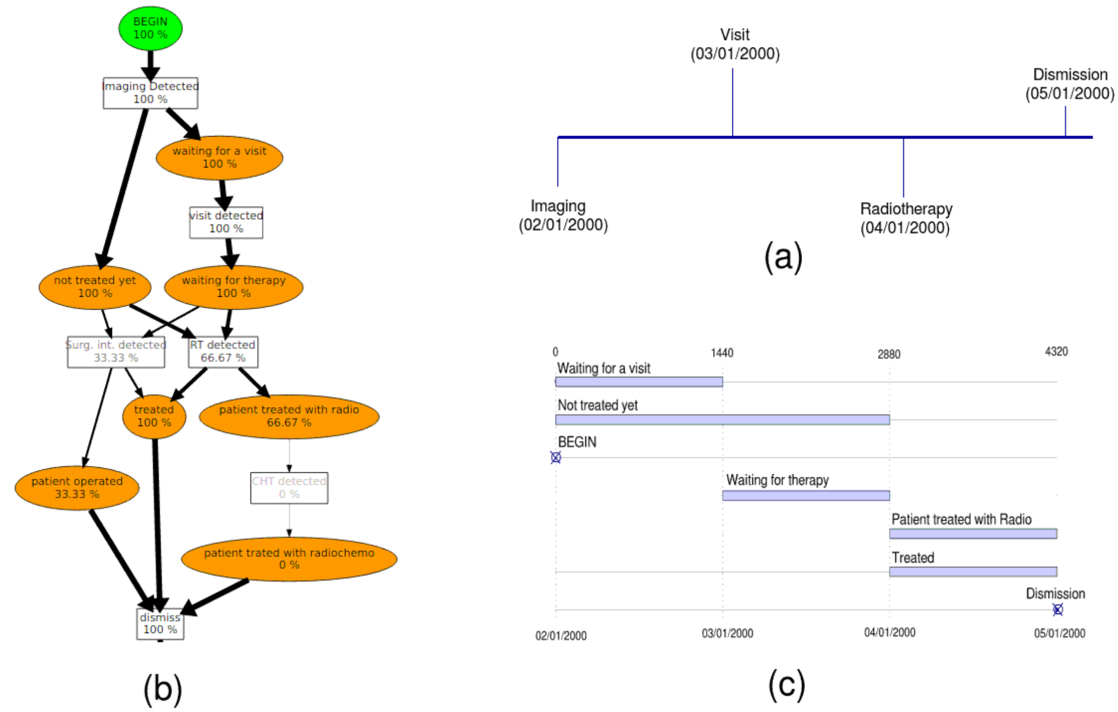

Fig. 2. (a) an example of track from an Event Log; (b) an example of a diagram generated using the PWL language. Round nodes correspond to states, boxes are triggers. Thicker arcs correspond to larger number of patients following the corresponding "path". Finally, (c) provides an example of the output for the track in (a): the result is a new track which can represent the compliance with the guideline of the process followed by a given patient, or a new event log for further investigation.

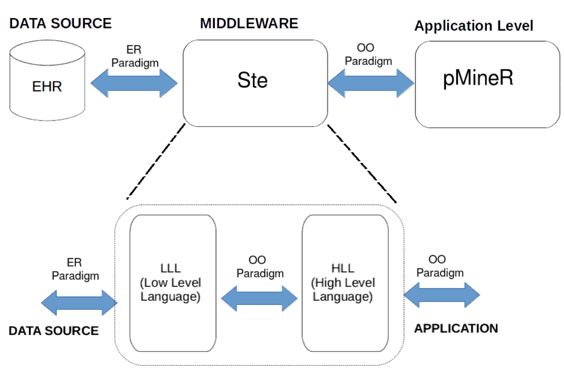

Fig. 3. An overview of Ste in a computational pipeline with pMineR. The lower part of the image shows the two-layers architecture of CSL, that is exploited by Ste. LLL maps the items of data presented following the ER paradigm, into suitable classes and objects of the object-oriented paradigm. HLL manipulates the inherited set of classes, and provides to the application level classes and methods closer to the actual needs.

LLL is playing the role of an Object Relational Mapping language (ORM). Common ORMs are, for example, Doctrine for PHP[5] or Hibernate[6] for Java. At the upper layer, the language HLL (Higher Level Language) allows to manipulate the classes inherited from the lower layer to enrich the availability of classes and methods (both in terms of numbers and abstractions) and present exploitable data to the user or to the exploited PM tool.

Figure 3 shows how Ste can be exploited for bridging the gap between pMineR and EHRs, and gives an overview of the interactions between the two CSL layers. The same approach can be adopted for not-ER data sources, for example csv files or SPARQL endpoint.

Remarkably, Ste has some similarities for example with D2RQ[7], an RDB-to-RDF mapping language [26]. D2RQ maps an ER schema to a triple store schema, providing a SPARQL Endpoint to the user. When a SPARQL query is submitted, D2RQ automatically produces, in real time, the needed SQL queries, builds the expected result according with the new paradigm and returns the result of the SPARQL query to the users. The mechanism of Ste is similar (and an example is provided in Figure 4), but the final representation is provided in an object-oriented syntax language, and is natively integrated in the R environment.

A. Event Log Generation

The package Ste, beside implementing and supporting the CSL language, provides a range of techniques for building

[5]http://www.doctrine-project.org/
[6]http://hibernate.org/orm/
[7]http://d2rq.org/

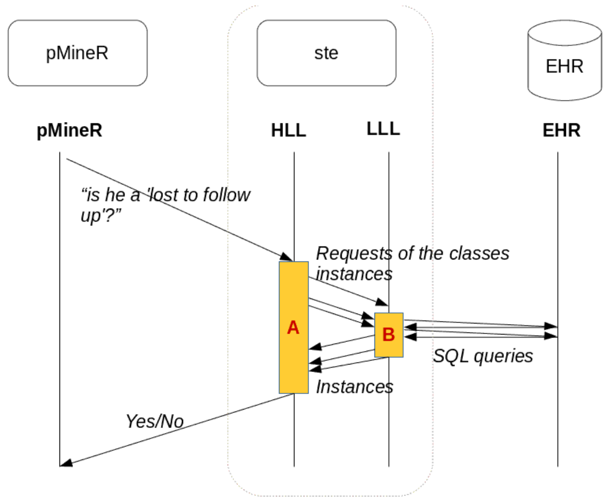

Fig. 4. An example of computation; the application, in this case pMineR, makes a query using the set of classes and methods made available by the HLL interface. This sort of queries usually require data related to a specific patients or to a set of patients sharing some common characteristics. HLL then queries LLL in order to gather the required data. LLL is then in charge of translating the requests received from HLL to SQL queries, pulls information from the EHR database, and returns the appropriate results. Eventually, the application receives the required data, without the need to cope with the actual EHR architecture.

an event log. The easiest approach is to adopt the *harvester* method, that receives as input one or more scripts written in HLL and LLL (necessary to link the data sources and to provide a description of involved concepts) and an XML file that contains information related to columns and rows of the event log that should be generated.

The **XML.Harvester.Example.xml** file included in the example archive has been encoded according to the ER schema shown in Figure 5 (part a). The same figure (part b) presents a subset of the associated UML class diagram, that is implemented via LLL and HLL. Most of the labels are in Italian, as the files correspond to those currently exploited in the Italian Gemelli Hospital. We provide English translations of relevant labels in the text. In the **XML.Harvester.Example.xml** XML file, the tag `row` indicates the rule to follow select appropriate rows: in the example, the Patient's ID should be selected by considering the explicitly passed values `187`, `188`, and `189`. Patients are then retrieved using the HLL method `engaged_in` on the class `Paziente` (patient). The tag `section` describes the columns of the EHR that should be considered as attributes of the instance of the class `Paziente`. The next three attributes `dataEvento` (event date), `descrizione` (description), `dimensioneTiroide` (thyroid size) are retrieved using a specified method, `hasClinicalEvents`, that maps the *1:n* relation between a specific instance of the class Patient and the Clinical events –specified as `columns` of the section in the XML– in the clinical history of the considered patient. The last two columns, `descrizione` (description of the exam), and `valore` (value), similarly, refers to a method able to map the *1:n* relation between a specific clinical event and the associated exams. The event log built following the described XML schema will have, for each patient, a number of rows that depends on the combinatorial explosion of the two *1:n* relations.

As an alternative to the harvester, Ste allows the creation of event logs via a direct access to the internal CSL engine: of course, this approach is more complex, and reduces part of the benefits of using Ste as a middle layer between the application and the EHR, but it allows to cope with a wider spectre of configurations and with more sophisticated structures or queries.

V. Conformance Checking with Ste

We are now in the position of exploiting Ste for performing conformance checking. We implemented in CSL, via Ste, a clinical guideline related to the treatment of thyroid diseases. We then used Ste to link an EHR of an Italian Centre specialized in thyroid diseases. The EHR includes more than $12,000$ patients, and the guidelines concern the execution of the Fine Needle Aspiration Cytology under the suspect of thyroid cancer [27] after an Ultrasound examination. To implement the clinical guidelines, the tables represented in the ER schema shown in Figure 5 have been mapped in the CSL language. The implementation includes:

- an LLL script (125 rows) to describe 5 Classes (Patients, ClinicalEvent, UltraSoundExamination, LaboratoryExam, Nodule) and 12 different methods for extracting data from the EHR;
- an HLL script (400 rows) to create 3 new methods for the class Patient and one new method for the class UltraSoundExamination;

Both the LLL and HLL scripts are available in the example archive (files **guideline.FNAC.LLL.txt** and **guideline.FNAC.HLL.txt**). The pivotal method of the class Patient is `isFNACIndicated`. From the provided code, it is easy to observe that this method is in charge of the whole conformance checking process. Given the ID of a patient, the method identifies the tasks to execute, and may therefore interact with the lower layer to collect data. The `isFNACIndicated` can be invoked using a syntax like:

```
Patient(< id >).isFNACIndicated.
```

The output of the computation performed by the scripts, that is also the overall output of the Ste component, comes under the form of a text file that summarises the execution, and can be optimised by modifying the `isFNACIndicated` method. It gives details about the path followed by each patient, and provides a fruitful support for investigations and analysis. It can also be used to generate graphical representations. In our analysis we wrote a dashboard in `Shiny`[8], a commonly exploited R component. The dashboard allows the user to (i) select a single patient from the EHR, and (ii) observe how the clinical pathway of the patient flows through the implemented

[8]https://shiny.rstudio.com

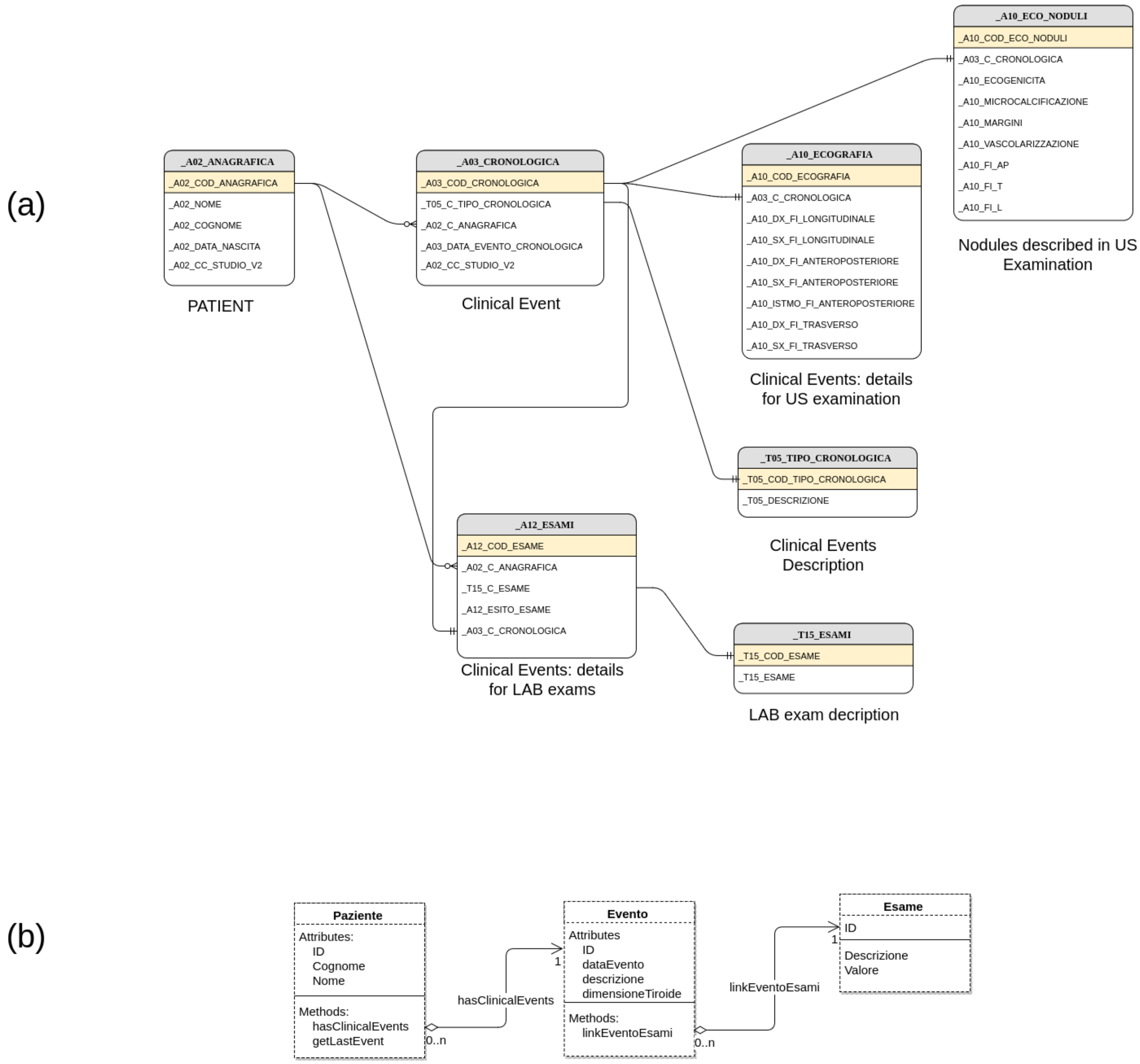

Fig. 5. (a) part of the Entity Relation schema of the considered EHR for patients treated for thyroid diseases. The organization of the tables reflects the needs of the architecture of the EHR, not the data representation issues. (b) a subset of the UML class diagram built using LLL and HLL. `Paziente` (patient) is a class referred to the Entity `Paziente`, the class `Evento` (clinical event) includes data from Entities `_A03_CRONOLOGICA`, `_A10_ECOGRAFIA`,`_T05_TIPO_CRONOLOGICA`, that correspond to three considered clinical events. The class `Esame` (laboratory exam) is a composition of Entities `_A12_ESAMI` and *_T15_ESAMI*, that represent two different types of lab exams. As correctly represented by the relations of the UML class diagram, a patient can have more clinical events, and for each clinical event there are some (or none) laboratory exams.

guideline. Evidently, such a tool can be a powerful instrument to identify critical patients, or to investigate overall deviations from the desired path described in the clinical guidelines. A snapshot of the dashboard is shown in Figure 6, and an interactive example is available at: http://5.249.147.20:7775/.[9] Using the provided interactive example, it is possible to analyse and investigate the path of the "test" patient. Other patients have been removed for privacy reasons, but the clinical guidelines are correctly implemented following Pacini et al. [27].

VI. Discussion and Conclusions

Process mining is gaining importance in the healthcare domain, also in the light of its ability to compare actually exploited medical processes –that can show aspects of personalized medicine– with clinical guidelines designed by considering the principles of evidence-based medicine. However,

[9]Please make sure that your firewall is not preventing connections to port 7775.

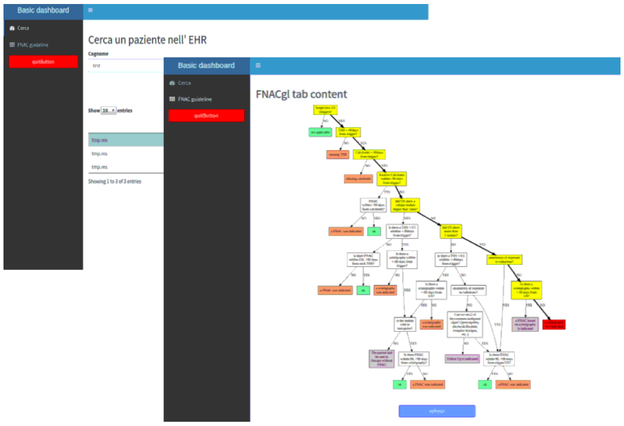

Fig. 6. The developed dashboard (currently, in Italian). By selecting a patient, it is possible to see the projection of her clinical pathway on the guidelines. The overall guidelines are shown, and the path followed by the patient is highlighted in yellow. Red is used to indicate the end point of the process.

the application of PM in the medical domain poses some significant challenges. One of the most significant issues is the lack of general approaches and tools for extracting data from existing and heterogeneous EHRs: therefore data and event logs are hard to gather, and can not be analysed in real-time. Existing tools for gathering data are very specific, are not well-integrated with PM pipelines and techniques, and require a significant pre-processing of data. Considering a different –but complementary– perspective, there are very few languages and approaches that allow to represent clinical guidelines and workflows, despite the fact that guidelines and workflows are necessary for performing conformance checking.

In this paper we proposed Ste, an R package that has been designed to support the integration in the PM pipeline of pMineR. Ste, levering on CSL, allows to automatise the generation of event logs by providing a useful mean for interacting with the existing EHR structure. CSL exploits the flexibility given by two layers, that provide, in turn, an interface for pMineR –or other PM tools able to deal with data provided under the form of object-oriented structures– and for the EHR. In fact, we show that CSL can also be used for encoding clinical guidelines, and is therefore a valuable support for performing conformance checking. We described the exploitation of Ste, in the larger architecture including also pMineR, for performing conformance checking on $12,000$ patients treated by a large Italian hospital. Such investigation lead to the identification of a number of remarkable aspects related to the use of Ste and pMineR.

- Ste showed to be effectively capable of generating Event Logs, when required by the architecture. The exploited approach, that relies on the design of HLL and LLL scripts is much more flexible and principled than the usual development of ad-hoc scripts. As a matter of fact, Ste also supports the reusability of the code, and points into the direction of a more principled approach to data gathering and generation of Event Logs. It is also easy to see how the use of CSL can lead to a number of available off-the-shelf scripts and methods that can be exploited (and configured) for a specific EHRs, in order to foster the exploitation of PM approaches and techniques.
- CSL provides a better support for temporal reasoning than PWL, and is therefore a better approach for dealing with complex guidelines. However, PWL allows to easily encode guidelines, as it is based on 3 constructs only, and can be inspected also by medical experts without additional help or support. Furthermore, PWL has been designed to natively support a graphical representation in the form of directed graphs, that can be straightforwardly exploited by medical experts. In fact, Ste can also be linked to R libraries for graphically representing flows and clinical guidelines –as shown in Figure 6–, but the implementation is left to the user, that has to carefully craft HLL and LLL scripts in order to obtain the suitable output.
- The use in the every day medical routine of Ste pointed out a promising way for combining Ste and PWL. Instead of invoking Ste only before the actual execution of pMineR, as for generating required event logs, it can be exploited online. Ste could be invoked, for instance, by some triggers specified in PWL: in this way Ste would timely generate online, during the execution of pMineR, only the actual data that need to be checked and analysed. This would allow to start from a minimal event log, and to query the EHR only in cases where additional information is needed.

We see several avenues for future work. We plan to extend the evaluation of the pMineR and Ste architecture with different data sets, in order to investigate how the approach can generalise on different sets of guidelines and events. Preliminary results in this direction indicate that Ste can easily deal with very complex guidelines, such as those designed for patients under treatment for a number of types of cancer.

Ste has been designed in order to extract data from EHRs, but there are sources of data that are not in EHR, and not all the EHRs grant access to the database schema. On this matter, LLL can be easily extended to deal, for instance, with SPARQL Endpoint and Pictura Archiving Communication System (PACS). We are interested in developing such extensions and testing the system on heterogeneous sources of data. Finally, we aim at providing an approach for supporting the design of graphical representations of guidelines and paths in Ste, in order to improve the usability of the tool.

References

[1] L. Wen-Ling and T. Fuu-Jen, “Personalized medicine: A paradigm shift in healthcare,” *BioMedicine*, vol. 3, no. 2, pp. 66 – 72, 2013.

[2] F. Collins, *The language of life: DNA and the revolution in personalised medicine*. Profile Books, 2010.

[3] D. W. Bates, G. J. Kuperman, S. Wang, T. Gandhi, A. Kittler, L. Volk, C. Spurr, R. Khorasani, M. Tanasijevic, and B. Middleton, “Ten commandments for effective clinical decision support: Making the practice of evidence-based medicine a reality,” *Journal of the American Medical Informatics Association*, vol. 10, no. 6, pp. 523–530, 2003.

[4] W. Aalst and et al, “Process mining manifesto,” in *Business Process Management Workshops*, 2011, pp. 169–194.

[5] R. Mans, W. M. P. van der Aalst, and R. J. B. Vanwersch, *Process Mining in Healthcare - Evaluating and Exploiting Operational Healthcare Processes*. Springer, 2015.

[6] R. Gatta, J. Lenkowicz, M. Vallati, E. Rojas, A. Damiani, L. Sacchi, B. De Bari, A. Dagliati, C. Fernandez-Llatas, M. Montesi, A. Marchetti, M. Castellano, and V. Valentini, *pMineR: An Innovative R Library for Performing Process Mining in Medicine*, 2017, pp. 351–355.

[7] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, 2016. [Online]. Available: R-project.org

[8] R. Gatta, M. Vallati, J. Lenkowicz, E. Rojas, A. Damiani, L. Sacchi, B. D. Bari, A. Dagliati, C. Fernández-Llatas, M. Montesi, A. Marchetti, M. Castellano, and V. Valentini, "Generating and comparing knowledge graphs of medical processes using pminer," in *Proceedings of the Knowledge Capture Conference, K-CAP*, 2017, pp. 36:1–36:4.

[9] R. Gatta, M. Vallati, C. Cappelli, B. D. Bari, M. Salvetti, S. Finardi, M. Muiesan, V. Valentini, and M. Castellano, "Bridging the gap between knowledge representation and electronic health records," in *HEALTHINF*, 2016, pp. 159–165.

[10] A. P. Kurniati, O. Johnson, D. Hogg, and G. Hall, "Process mining in oncology: A literature review," in *Information Communication and Management (ICICM), International Conference on*. IEEE, 2016, pp. 291–297.

[11] C. Fernandez-Llatas, A. Lizondo, E. Monton, J.-M. Benedi, and V. Traver, "Process mining methodology for health process tracking using real-time indoor location systems," *Sensors*, vol. 15, no. 12, pp. 29 821–29 840, 2015.

[12] E. Rojas, J. Munoz-Gama, M. Sepúlveda, and D. Capurro, "Process mining in healthcare: A literature review," *J Biomed Inform.*, vol. 61, pp. 224–s36, 2016.

[13] A. Kurniati, O. Johnson, D. Hogg, and G. Hall, "Process mining in oncology: A literature review," in *Proceedings of the 6th International Conference on Information Communication and Management ICICM 2016*, 2016, pp. 291–297.

[14] G. Hripcsak, "Writing arden syntax medical logic modules," *Computers in Biology and Medicine*, vol. 24, no. 5, pp. 331–363, 1994.

[15] Y. Shahar, S. Miksch, and P. Johnson, "The asgaard project: A task-specific framework for the application and critiquing of time-oriented clinical guidelines," *Artificial Intelligence in Medicine*, vol. 14, pp. 29–51, 1998.

[16] M. Peleg, A. Boxwala, E. Bernstam, S. Tu, R. Greenes, and E. Shortliffe, "Sharable representation of clinical guidelines in glif: relationship to the arden syntax," *J Biomed Inform.*, vol. 34, no. 3, pp. 170–81, 2001.

[17] M. Peleg, "Computer-interpretable clinical guidelines: A methodological review," *Journal of Biomedical Informatics*, vol. 46, no. 4, pp. 744–763, 2013.

[18] B. F. Van Dongen, A. K. A. de Medeiros, H. Verbeek, A. Weijters, and W. M. Van Der Aalst, "The prom framework: A new era in process mining tool support," in *International Conference on Application and Theory of Petri Nets*, 2005, pp. 444–454.

[19] C. W. Günther and A. Rozinat, "Disco: Discover your processes." *BPM (Demos)*, vol. 940, pp. 40–44, 2012.

[20] C. Fernández-Llatas, J. Benedi, J. García-Gómez, and V. Traver, "Process mining for individualized behavior modeling using wireless tracking in nursing homes," pp. 15 434–15 451, 2013.

[21] A. Dagliati, L. Sacchi, A. Zambelli, V. Tibollo, L. Pavesi, J. Holmes, and R. Bellazzi, "Temporal electronic phenotyping by mining careflows of breast cancer patients," *Journal of Biomedical Informatics*, vol. 66, pp. 136–147, 2017.

[22] C. M. Bishop, "Mixture models and EM," in *Pattern Recognition and Machine Learning*, 9th ed., ser. Information Science and Statistics. Springer, 2006, pp. 450–455.

[23] L. Kaufman and P. J. Rousseeuw, "Clustering by means of medoids," in *Statistical Data Analysis based on the L1 Norm*, 1987, pp. 405–416.

[24] M. Samwaldab, K. Fehrec, J. BruinaKlaus, and P. Adlassnigac, "The arden syntax standard for clinical decision support: Experiences and directions," *Journal of Biomedical Informatics*, vol. 45, no. 4, pp. 711–718, 2012.

[25] M. Sordo, A. Boxwala, O. Ogunyemi, and R. Greenes, "Description and status update on gello: a proposed standardized object-oriented expression language for clinical decision support." *Stud Health Technol Inform.*, vol. 107, pp. 164–8, 2004.

[26] M. Hert, G. Reif, and H. C. Gall, "A comparison of rdb-to-rdf mapping languages," in *Proceeding I-Semantics '11 Proceedings of the 7th International Conference on Semantic Systems*, 2011, pp. 25–32.

[27] F. Pacini, M. Schlumberger, H. Dralle, R. Elisei, J. W. Smit, and W. Wiersinga, "European consensus for the management of patients with differentiated thyroid carcinoma of the follicular epithelium. european thyroid cancer taskforce," *Eur J Endocrinol*, vol. 154, no. 6, pp. 787–803, 2006.